



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/712,384	11/12/2003	William John Gallagher	BEAS-01316US3	9603
23910	7590	01/04/2007	EXAMINER	
FLIESLER MEYER LLP 650 CALIFORNIA STREET 14TH FLOOR SAN FRANCISCO, CA 94108			NGUYEN, PHILLIP H	
			ART UNIT	PAPER NUMBER
			2191	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/04/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)
	10/712,384	GALLAGHER, WILLIAM JOHN
	Examiner Phillip H. Nguyen	Art Unit 2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 27 November 2006.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-24 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-24 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 12 November 2003 is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date. _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the amendment filing of 11/27/2006. Claims 1-24 are pending and have been considered below.

Specification

2. The use of the trademark JAVA™ has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner, which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claim 23 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 23 is non-statutory because it recites a computer program product, which is directed to software, per se, lacking storage on a medium, which enables any underlying functionality to occur. Software is descriptive material, per se, and is therefore non-statutory.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claim 22 recites the limitation "wherein the program code for determining whether a resource is available includes program code for" is unclear to the examiner. This limitation was deleted from claim 1. There is insufficient antecedent basis for this limitation in the claim. For the examining purposes, examiner interprets "determining whether a remote object having an interface to which code is being written is available" as further limited of claim 1. Further more, the language of claim 22 also raises a question as to whether the generating program code step still performs if the code being written is not available. If the code is not available then the generating program code step never performs. This defeats the purpose of code generation.

Appropriate correction is required.

Response to Amendment

7. Per Applicant's arguments filed on 11/27/2006 have been fully considered but they are moot in view of the new grounds of rejection.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

Art Unit: 2191

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claims 1-10, 12, 14-18, and 22-24 are rejected under 35 U.S.C. 102(e) as being anticipated by Boehme et al. (United States Patent No.: US 6,578,191 B1).

As per claim 1:

Boehme discloses a method for dynamically generating program code, comprising:

- dynamically generating program code ("dynamically generate and load adapter class" Col 2, line 62), wherein dynamically generating program code includes:
 - o creating a class file container object ("Adapter classes and objects are automatically and dynamically generated" Col 3, line 2-3; "ClassInfo newClass=new ClassInfor ..." Col 4, line 30-33);
 - o adding a method to the class file object ("newClass.addMethod(ACC_PUBLIC, "actionPerformed", "(LactionEvent;)V", bytecodes));
 - o adding code to the method using programming language constructs ("code is created for the adapter class initialization methods" Col 5, line 49-50; "code is created for the adapter class methods referenced in the adapter class specification" Col 5, line 52-53);

Art Unit: 2191

- o generating byte code for the class file container object ("bytecodes necessary to construct the adapter class, interface, field, methods, and attributes are generated" Col 4, line 24-26); and
- o instantiating an instance of the new class file object ("An instance of the adapter class is instantiated in function block 107" Col 4, line 56).

As per claim 2:

Boehme discloses the method as in claim 1 above; and further discloses wherein creating a class file container object includes:

- setting attributes for a class file
("newClass.setSuperClassName("com/ibm/bml/EventAdapterImpl");" Col 4, line 35, setting the super class name).

As per claim 3:

Boehme discloses the method as in claim 2 above; and further discloses wherein the attributes include:

- a class file name
("newClass.setSourceFilename("BMLActionEventAdapter")" Col 4, line 32); and

Art Unit: 2191

- a parent super class

(“**newClass.setSuperClassName(“com/ibm/bml/EventAdapterImpl”);**” Col 4, line 35).

As per claim 4:

Boehme discloses the method as in claim 1 above; and further discloses wherein adding a method to the class file object includes:

- adding a plurality of methods to the class file object (“**the bytecodes necessary to construct... methods**” Col 4, line 24-25; **newClass.addSpecialMethod(...); newClass.addMethod(...);**” Col 4, line 40-43).

As per claim 5:

Boehme discloses the method as in claim 1 above; and further discloses:

- the programming language constructs correspond to programming language statements, expressions, and variables (Col 4, line 10-20, **this method contains statements, expressions, and variables...**).

As per claim 6:

Boehme discloses the method as in claim 5 above; and further discloses wherein the programming language constructs include:

Art Unit: 2191

- parameters (“**beanInfo=Introspector.getBeanInfo(jugglerClass)**...” Col 4, line 10, **jugglerClass is a parameter**).

As per claim 7:

Boehme discloses the method as in claim 5 above; and further discloses:

- wherein each statement, expression type, and variable is represented as an object (for example, “**<bean class = “java.awt.button”>** Col 3, line 62, this is a statement contains an expression of bean class set equal to **java.awt.button object**;
- “**beanInfo=Introspector.getBeanInfo(jugglerClass)**” Col 4, line 10, **jugglerClass is a variable and represents an object**).

As per claim 8:

Boehme discloses the method as in claim 1 above; and further discloses wherein generating byte code for the class file container object includes:

- generating an intermediate representation of program flow (“**op-codes**” Col 4, line 27).

As per claim 9:

Boehme discloses the method as in claim 8 above; and further discloses wherein generating byte code for the class file container object includes:

- converting the intermediate representation into byte code ("the bytecodes necessary to construct the... are generated based upon the wiring description, and the op-codes" Col 4, line 24-28).

As per claim 10:

- Boehme discloses the method as in claim 1 above; and further discloses
- wherein the program code implements an adapter class ("to construct adapter class" Col 4, line 24-25; also see Col 4, line 30-55).

As per claim 12:

Boehme discloses the method as in claim 1 above; and further discloses program code for:

- repeatedly adding a method to the class file object for each method associated with a stub generated for a remote object (Col 4, line 40-43, it repeatedly adds methods to the class).

As per claim 14:

Boehme discloses the method as in claim 1 above; and further discloses wherein the program code for adding code to the method includes program code for:

- repeatedly adding code for each method added to the class file ("code is created for the adapter class initialization methods" Col 5, line 49-50; "code is created for the adapter class methods referenced in the adapter

Art Unit: 2191

class..." Col 5, line 52-53, which means, code is created repeatedly for each method in order to fulfill the functionality of the methods).

As per claim 15:

Boehme discloses the method as in claim 1 above; and further discloses wherein at least one of the program code for adding a method to the class file and the program code for adding code to the method includes program code for:

- generating a tree of statements (Col 4, line 10-20, **a tree represents at least one method containing at least one code statement, expression, variable, or other programming construct**).

As per claim 16:

Boehme discloses the method as in claim 15 above; and further discloses wherein the program code for generating a tree of statements includes program code for:

- generating a tree representing at least one method (Col 4, line 10-20, **is a method**), the at least one method comprising at least of code statement, an expression, a variable and a programming construct (Col 4, line 10-20, **this method contains statements, expressions, and variables**).

Art Unit: 2191

As per claim 17:

Boehme discloses the method as in claim 15 above; and further discloses wherein the program code for generating a tree of statements includes program code for:

- generating a tree forming a known structure when the class file container is a known type ("adapter type" Col 3, line 57).

As per claim 18:

Boehme discloses the method as in claim 17 above; and further discloses wherein the program code for generating a tree forming a known structure when the class file container is a known type includes program code for:

- generating a tree forming a known structure when the class file container is of at least one of an adapter and a proxy type ("adapter type" Col 3, line 57).

As per claim 22:

Boehme discloses the method as in claim 1 above, but does not explicitly disclose determining whether a remote object having an interface to which code is being written is available. It is inherent in Boehme that code is always available in order to perform code generation process.

Art Unit: 2191

As per claim 23:

Boehme discloses the method as in claim 1 above, but does not explicitly disclose the dynamically generated program code would be configured to exist for the life of a server the dynamically generated program code resides upon. It is inherent in Boehme's approach in order for the generated program code to be executed.

As per claim 24:

Boehme discloses the method as in claim 1 above, but does not explicitly disclose computer code for determining whether a resource is available. It is inherent in Boehme's approach in order to fulfill the purpose of code generation.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. Claims 11 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boehme et al. (United States Patent No.: US 6,578,191 B1), in view of Cohen et al. (United States Patent No.: 6,011,918).

As per claim 11:

Boehme discloses the method as in claim 1 above, but does not explicitly

Art Unit: 2191

disclose wherein the program code implements a proxy class.

However, Cohen discloses an analogous computer program product wherein the program code implements a proxy class ("proxy class is called X, ... will execute on the local machine" Col 15, line 32-34).

Therefore, it would have been obvious to one having an ordinary skill in the art to modify Boehme's approach to implement proxy class. One having an ordinary skill in the art would have been motivated to implement proxy class so that **each of the constructed methods make corresponding remote method calls using Java's RMI function** (see Cohen Col 14, line 5-10).

As per claim 13:

Boehme discloses the method as in claim 12 above, but does not explicitly disclose wherein the program code for repeatedly adding a method to the class file object for each method associated with a stub generated for a remote object includes program code for: determining a number of methods associated with the stub in a remote interface.

However, Cohen discloses an analogous computer program product for determining a number of calling methods ("the weighting of the relationships between the programmed methods is performed by determining for each calling programmed method" Col 3, line 52-55).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Boehme's approach to determine the

Art Unit: 2191

number of method added to the class file object. One having an ordinary skill in the art would have been motivated to determine the number of method by include a counter or an indicator that increment each time a method is adding to the class file object so it would know when it reaches the end of the adding iteration process.

12. Claims 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boehme et al. (United States Patent No.: US 6,578,191 B1), in view of Stapp et al. (United States Patent Application Publication No.: US 2004/0015832 A1).

As per claim 19:

Boehme discloses the method as in claim 1 above, but does not explicitly disclose wherein the program code for generating byte code for the class file container object includes program code for maintaining a state of a program being generated by each statement.

However, Stapp discloses an analogous computer program product that includes program code for maintaining a state of a program being generated by each statement ("Beans also have persistence, which is a mechanism for storing the state of a component in a safe place" Paragraph 0039).

Therefore, it would have been obvious to one having an ordinary skill in the art to modify Boehme's approach to maintaining a state of a program being generated. One having an ordinary skill in the art would have been motivated to modify Boehme's

Art Unit: 2191

approach because it allows a component (bean) to retrieve data that a particular user had already entered in an earlier user session (Paragraph 0039).

As per claim 20:

Boehme and Stapp disclose the method as in claim 19 above; and Stapp further discloses wherein the program code for maintaining a state of a program being generated by each statement includes program code for:

- maintaining at least one of a content of a stack, a content of a variable in use during program flow (“**a component (bean) to retrieve data that a particular user had already entered in an earlier user session**” Paragraph 0039, **data is contents of variables**).

As per claim 21:

Boehme and Stapp disclose the method as in claim 20 above; and Boehme further discloses:

- generating an intermediate representation of program flow based upon the at least one of a content of a stack, a content of a variable in use during program flow (“**the bytecodes necessary to construct the adapter class, interface, fields, methods, and attributes are generated...**” Col 4, line 24-29, **op-codes are intermediate representation of program flow can be found in bytecode, specifies operation to perform**).

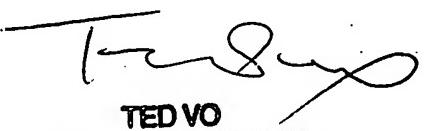
Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Friday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
12/13/2006


TED VO
PRIMARY EXAMINER
TECHNOLOGY CENTER 2100